

图像基本运算：几何变换

2019年11月9日 22:13

简介

图像的几何变换包括图像的镜像、缩放以及旋转，以下重点介绍图像的缩放和旋转

原理

(1) 图像缩放分为比例缩放和非比例缩放，比例缩放指在 x 方向和 y 方向按相同的比例缩放，否则为非比例缩放。

设图像在 x 方向上放大到 a 倍，在 y 方向上放大到 b 倍（若 $a = b$ 即为比例缩放，否则为非比例缩放），则原始图像中的 $A_0(x_0, y_0)$ 比例缩放后，与新图中的 $A_1(x_1, y_1)$ 的坐标对应关系为

$$\begin{cases} x_1 = ax_0 \\ y_1 = by_0 \end{cases}$$

(2) 图像旋转，一般情况下的图像旋转是指以图像的中心为原点，将图像上的所有像素都旋转同一个角度的变换。在图像旋转变换中，既可以把转出显示区域的图像截去，也可以扩大显示区域的图像范围已显示图像的全部。

设图像的任一点 $A_0(x_0, y_0)$ 经旋转角度 β 以后到新的位置 $A_1(x_1, y_1)$ ，则坐标对应关系为

$$\begin{cases} x_1 = x_0 \cos \beta + y_0 \sin \beta \\ y_1 = -x_0 \sin \beta + y_0 \cos \beta \end{cases}$$

(3) 灰度插值，若缩放或旋转所产生的图像中的像素在原图像中没有对应的像素点时，就需要进行灰度插值，常用的灰度插值方法有三种——最近邻法、双线性插值法和三次内插法。

MATLAB代码

```
%%
clc;clear;
image = imread('cameraman.tif');
[m, n] = size(image);
figure(1);imshow(image);title('原图像')

%%
%图像的缩放
imageZoom1 = imresize(image, 2);
figure(2);imshow(imageZoom1);title('放大两倍的图像（比例缩放）');

imageZoom2 = imresize(image, [m * 2, n]);
```

```

figure(3);imshow(imageZoom2);title('高放大两倍的图像（非比例缩放）');

imageZoom3 = imresize(image, 2, 'nearest');
figure(4);imshow(imageZoom3);title('放大两倍的图像（最近邻法）');

imageZoom4 = imresize(image, 2, 'bilinear');
figure(5);imshow(imageZoom4);title('放大两倍的图像（双线性插值）');

imageZoom5 = imresize(image, 2, 'bicubic');
figure(6);imshow(imageZoom5);title('放大两倍的图像（双三次插值）');

%%
%图像的旋转
imageRotate1 = imrotate(image, 30, 'crop');
figure(7);imshow(imageRotate1);title('旋转30度后剪裁');

imageRotate2 = imrotate(image, 30, 'loose');
figure(8);imshow(imageRotate2);title('旋转30度后不剪裁');

```

实验结果及分析

由于本篇讲解图像缩放及旋转，为了使图片见有更明显的区别，文档中的图片不调整大小，使用从MATLAB中复制图片的默认大小

原图像依旧是用cameraman.tif

原图像



下面是比例缩放和非比例缩放后的图像，需要指出的是，imresize函数默认使用双三次插值

放大两倍的图像（比例缩放）



高放大两倍的图像（非比例缩放）



下面三个依次使用最近邻法、双线性插值法、双三次插值法进行比例缩放

放大两倍的图像（最近邻法）



放大两倍的图像（双线性插值）



放大两倍的图像（双三次插值）



可以看出，相比于其他两种插值方法，最近邻法有比较明显的马赛克效应

图像旋转后剪裁和不剪裁的结果，需要指出imrotate函数默认使用最近邻法

旋转30度后剪裁



旋转30度后不剪裁

